

CS556 Introduction to Cryptography - Prof. Ming-Deh Huang
Scribe: Iftikhar A Burhanuddin
burhanud@usc.edu
Class #1 - August 27, 2002

Administrivia: Syllabus, Academic integrity, Announcements, Homework solutions at <http://www-rcf.usc.edu/~mdhuang/cs556>. Please check regularly!

Professor's Office: SAL 314 Tel: x04783 email: huang@pollux.usc.edu
Office hour: Th 11-12 (tentative)

Textbook:

1. A Course in Number Theory and Cryptography, Neal I. Koblitz, Graduate Texts in Mathematics, # 114, Springer, September 1994. *Main text. Thin and dense.*
2. Applied Cryptography, Bruce Schneier, John Wiley, October 1995. *Supplementary text, strongly recommended. Thick and user-friendly.*
3. Material posted on the course webpage. Papers, etc.

Course contents: Why and how cryptosystems work. The focus will be on foundations of cryptography and public key cryptosystems like:

1. Integer factoring based RSA cryptosystem
2. Discrete logarithm based schemes like Diffie Hellman, El Gamal
3. Elliptic curve and lattice based cryptosystems (if time permits)

We'll also talk about the recent breakthrough result on a deterministic polynomial time algorithm for primality testing. The course will not talk about security and/or applications - digital cache, cyber security, etc.

Grading: Assignment of weights are subject to change. HW: 20%, Midterm: 40%, Final: 40%. Homeworks may include computational assignments.

Prerequisites: CS 570 - Analysis of Algorithms or Instructor's approval

Lecture

Cryptography: is the study of methods of encryption and decryption. *Encryption* is disguising a message (aka *plaintext*) so that only the intended recipient is able to read the message. *Decryption* is unravelling the encrypted message (aka *ciphertext*).

We chop the message into smaller sized message units. A natural message unit can be a letter/character, pair of letters (digraph), block of letters, etc.

Say \mathbf{P} is the set of all plaintext message units and \mathbf{C} is the set of all ciphertext message units. Hence encryption can be viewed as a 1-1 function between \mathbf{P} and \mathbf{C} . So cryptography is in some sense the science of developing good functions.

The earliest known example of a cipher is called the Caesar cipher as it was believed to be used by Caesar. It works as follows:

1. Label the letters of the English alphabet $\{A, B, \dots, Z\}$ by numbers:
 $A \rightarrow 0, B \rightarrow 1, C \rightarrow 2, \dots$
2. Choose a secret number k called the *key* $\in \{0, 1, \dots, 25\}$
3. To encrypt a letter push it forward by k positions, so $A \rightarrow D, B \rightarrow E, C \rightarrow F, \dots, Z \rightarrow C$
4. To decrypt shift the letter k positions backward

That is for $x \in \{0, 1, \dots, 25\}$

$$E(x) = \begin{cases} x + k & \text{if } x < 26 - k \\ x + k - 26 & \text{if } x \geq 26 - k \end{cases}$$

A more succinct representation is $E(x) = x + k \pmod{26}$ and $D(x) = x - k \pmod{26}$. Also note that $D(E(x)) = x$! If $k = 3$ then YES would be encrypted as BHU

$$\begin{array}{ccccccc} Y & E & S & \longrightarrow & B & H & U \\ 24 & 4 & 18 & & 1 & 7 & 21 \end{array}$$

This scheme is very easy to break as there are only 26 possibilities. How can we improve on the Caesar cipher? We could encrypt pairs of letters at a time (digraphs) $\{AA, AB, \dots\}$ and now the number of possible keys are 26^2 .

We could also encrypt blocks of 3, 4, 5, . . . letters. But as we increase the size of the sets \mathbf{P} and \mathbf{C} encrypting and decrypting become harder in this setting. Also it takes only one plaintext and its corresponding ciphertext to recover the key and break the system. So in this scheme the security of the system depends on the security of the key.

In the Caesar cipher the encryption and decryption keys k happen to be the same but in general the keys may be different. If the decryption key is derivable from the encryption key the scheme is called *Symmetric Key*. On the other hand if the decryption key is “not easy”¹ to derive from the encryption key or vice-versa it’s called *Asymmetric Key Scheme* aka *Public Key Cryptography*.

Anecdote for the day: Once upon a time in the court of a king, the hair of a messenger would be shaved off and a key branded on his head. When the messenger’s hair grew back he would be sent to the battlefield carrying the hidden key. The messenger would lose his hair again and the general would get his key. *Moral of story: Key management is painful!*

¹ “not easy” will be defined complexity theoretically in the coming classes.